This guide describes the steps that are needed to be able to compile NML NewGRFs on a Windows system using the make system as used by the #openttdcoop DevZone. This guide is done on a clean install of Windows 7, some names might be different on your system if you use a different version.

Programs
To compile NML NewGRFs on a Windows system, the following programs are needed:

- NML
- TortoiseHg
- Python (version 2.7.x)
- MinGW/MSYS

All programs in this guide will be installed in folders on the C:\-drive itself, so not in Program Files or somewhere else. If you install programs in a different location, you have to update the paths in later steps. DO NOT USE FOLDERS WITH WHITESPACES IN THEM. All programs will be added to the %PATH% variable, which will be done at the end.

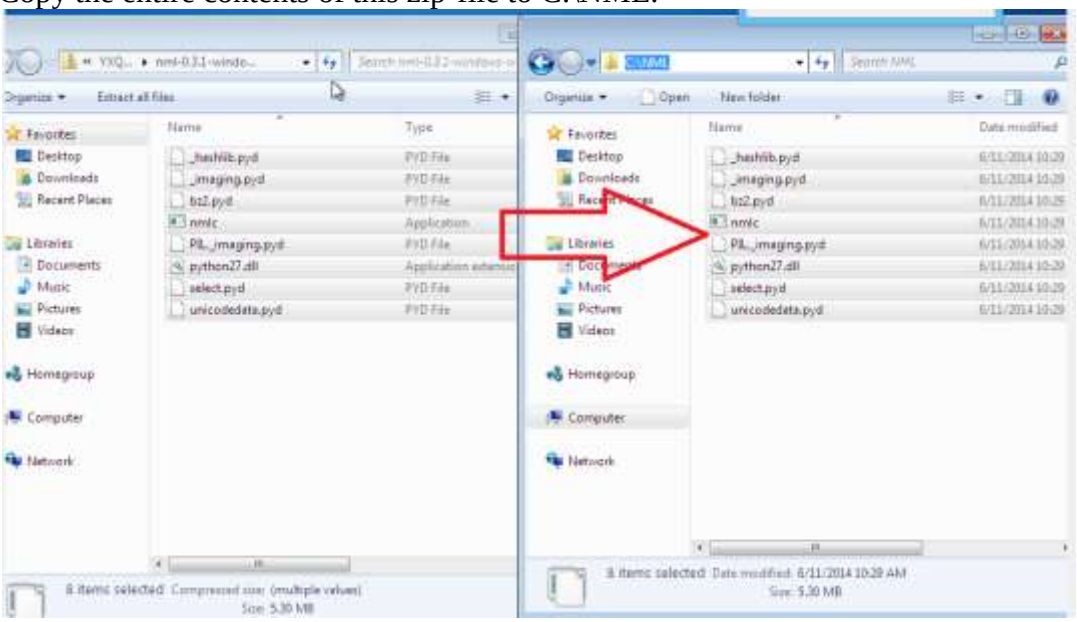
NML
NML can be downloaded from http://bundles.openttdcoop.org/nml/ . You can choose between Releases and Push. Releases are marked as stable and also contain older versions if needed, while Push contains the latest changes. For this guide it does not matter which one you use, the steps are the same for all versions.

Download the file named nml-<version>-windows-win32.zip, where <version> is the version number of NML you are downloading. The other files are not relevant.

Index of /nml/releases/LATEST

Name	Last modified	Size
Parent Directory		-
docs/	10-May-2014 09:49	-
zpl-2.1.txt	10-May-2014 09:49	2.0K
readme.txt	10-May-2014 09:49	4.6K
license.txt	10-May-2014 09:49	18K
changelog.txt	10-May-2014 09:49	14K
release.txt	10-May-2014 09:49	72
nml_notepadpp.xml	10-May-2014 09:49	32K
nml_kate.xml	10-May-2014 09:49	49K
nml-0.3.1.r5242_f6a3ae1163ab-py2.7.egg.md5	10-May-2014 09:49	73
nml-0.3.1.r5242-f6a3ae1163ab.tar.gz.md5	10-May-2014 09:49	70
nml-0.3.1.r5242-f6a3ae1163ab.linux-x86_64.tar.gz.md5	10-May-2014 09:49	83
nml-0.3.1-windows-win32.zip.md5	10-May-2014 09:49	62
nml-0.3.1.r5242-f6a3ae1163ab.tar.gz	10-May-2014 09:49	357K
nml-0.3.1.r5242_f6a3ae1163ab-py2.7.egg	10-May-2014 09:49	573K
nml-0.3.1.r5242-f6a3ae1163ab.linux-x86_64.tar.gz	10-May-2014 09:49	428K
nml-0.3.1-windows-win32.zip	10-May-2014 09:49	2.8M

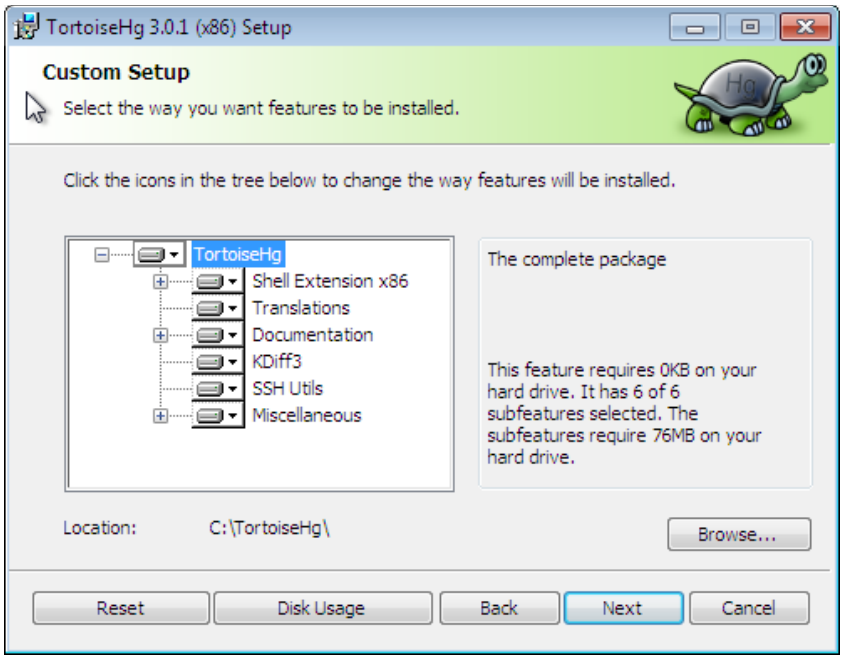
Copy the entire contents of this zip-file to C:\NML.



TortoiseHg

TortoiseHg is revision control client that is used to pull the code and changes from the #openttdcoop-server and push your own changes back. The download button can be found at http://tortoisehg.bitbucket.org

In the installation, do not forget to change the location (C:\TortoiseHg\ in this guide). If the Browse… button is greyed out, make sure you have selected TortoiseHg in the feature tree. The other settings are fine.



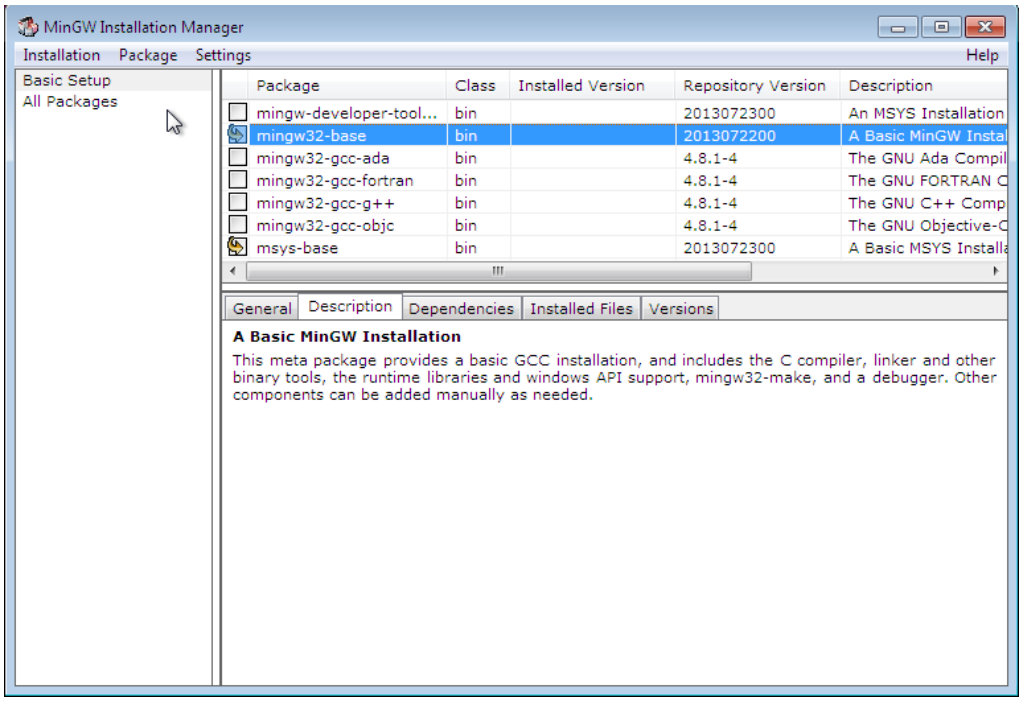
Python

Python can be downloaded from https://www.python.org/downloads/releases/2.7.7/ . Choose the installer that matches your system, or stick with the x86-version if you are unsure fo which version to use. The installer will default to install in C:\Python27\ . In the customization window you can opt to already add python.exe to your path. You can let the installer do it for you, or do it yourself when you add the other programs, both are fine. In this guide it will be done manually.

MinGW/MSYS
MinGW is the magic that adds the make and other commands you need to your Windows installation. The installer can be downloaded from http://sourceforge.net/projects/mingw/files/Installer/ .

In the first window you can set the install location and some other settings. By default it installs in C:\MinGW\. There is no need to install program shortcuts in the start menu and on the desktop, as you will only use the command line to issue the make command, so you can deselect those options. After this, it will download some stuff and then open the MinGW Installation Manager.

You need to select at least mingw32-base and msys-base. If you need more packages at a later moment, you can always run the MinGW Installation Manager again. After selecting the packages you want, you can apply them through the Apply Changes option in the Installation menu. Once it has applied the changes, you can close the Installation Manager.



The Path-variable

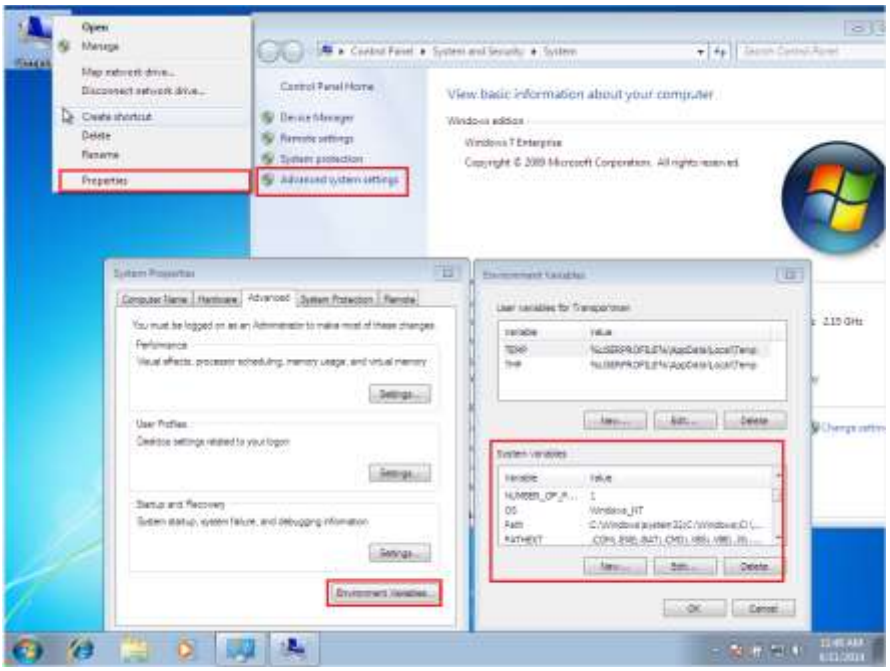
All the needed programs are now installed on your computer and work, however, calling the command from the command line will likely result in "'Program' is not recognized as an internal or external command, operable program or batch file."

To have the command line recognize the commands you need, you need to add the folders that contain the programs to the Path-variable. Changing the Path-variable might break your computer if you remove something that you should not remove. If you follow the steps precisely, nothing will happen.

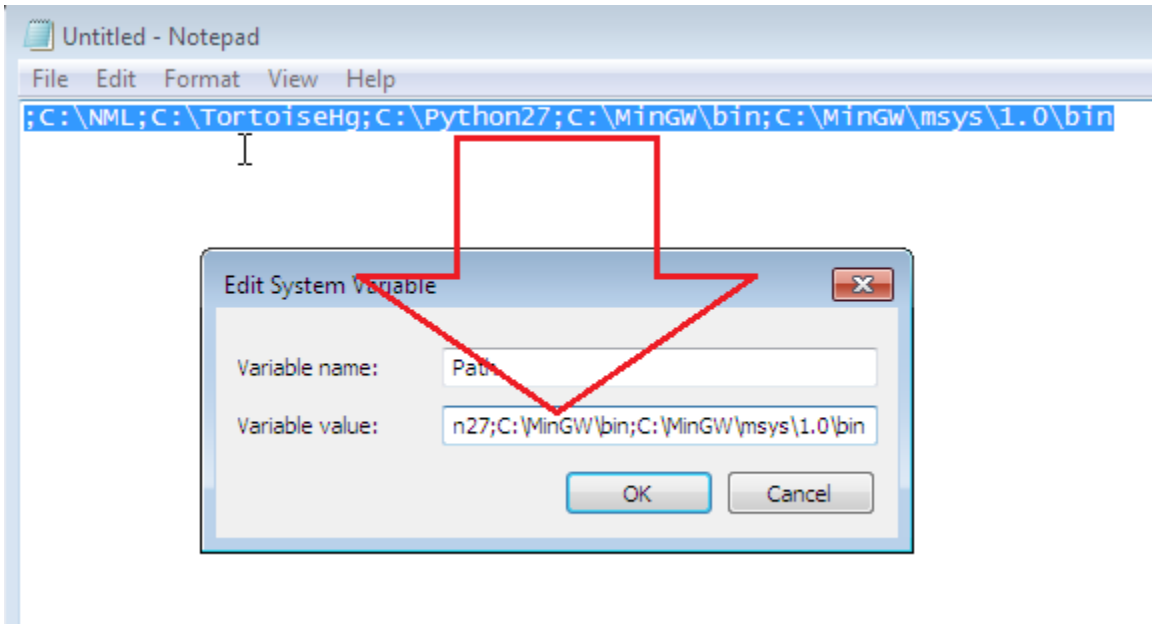
First, open Notepad or another text editor of your liking. If you installed the programs directly in a folder on C:\, you can copy the following line:
;C:\NML;C:\TortoiseHg;C:\Python27;C:\MinGW\bin;C:\MinGW\msys\1.0\bin

If you installed in a different location, you need to change the paths to point to the correct location. For NML, TortoiseHg and Python that is just the install directory, for MinGW you need the \bin subdirectory in the install directory and for MSYS you need the \msys\1.0\bin subdirectory of the MinGW install directory.

The easiest way to access the System variables is using the following steps:
Right click on Computer and select properties
In the window that opens, choose System Properties
On the Advanced tab, select Environment Variables
The Path variable is under System variables



Select the Path-variable and choose Edit…
Add the whole line from your text editor with folders to the end of the Value, without touching anything else.



Press OK until all windows are closed.

To check if everything works, open cmd.exe and issue the following commands:
nmlc --version
hg
python
exit()

bash
exit

If all commands work properly, the programs are installed correctly. If one of the commands failed, check if the program is installed, run the command from the installation folder for that program (use the cd [path] command to change folders while on the command line) and check if you changed the Path-variable correctly.

Compiling your NewGRF

If all is working up to now, you can change the directory using the cd command to the folder containing your code. You can then use the "make" command to start the compiling of your NewGRF.

Common errors

Command not found in bash

If you get errors that certain commands are not found in bash, it might be that the path in bash is not set correctly. You can check this by starting the command prompt (cmd.exe) and issue the following commands two commands:
bash
echo $PATH

It should list all the paths you added earlier to your Windows Path-variable, although the style will be different (paths start with /c/ instead of C:/ and different paths are separated by a : instead of a ;). If the list does not contain all the paths you added earlier, that might be the cause of the problem.

To fix that, you need to modify/create a file called .bashrc in C:\MinGW\msys\1.0\home\<username>\, but Windows does not allow creation of files starting with a . using Windows Explorer. You need the command line for this. cd to C:\MinGW\msys\1.0\home\<username>\ and use the following command to create the file:
echo.>.bashrc

Open this file using notepad or any other text editor and add the following text to it:
PATH="/c/NML/:/c/TortoiseHg:/c/MinGW/bin/:/c/MinGW/msys/1.0/bin:/c/Python:$PATH"

Save the file and try compiling your NewGRF again.